Lighting-grounded Video Generation with Renderer-based Agent Reasoning

Ziqi Cai[1,2,4] Taoyu Yang[1,2] Zheng Chang[5] Si Li[5] Han Jiang[4] Shuchen Weng[3,1] Boxin Shi[1,2,*]

[1]State Key Laboratory for Multimedia Information Processing, School of Computer Science, Peking University
[2]National Engineering Research Center of Visual Technology, School of Computer Science, Peking University
[3]Beijing Academy of Artificial Intelligence [4]OpenBayes Information Technology Co., Ltd.
[5]School of Artificial Intelligence, Beijing University of Posts and Telecommunications

{czq,yangty1031}@stu.pku.edu.cn, {zhengchang98,lisi}@bupt.edu.cn,
hahn@openbayes.com, {shuchenweng,shiboxin}@pku.edu.cn

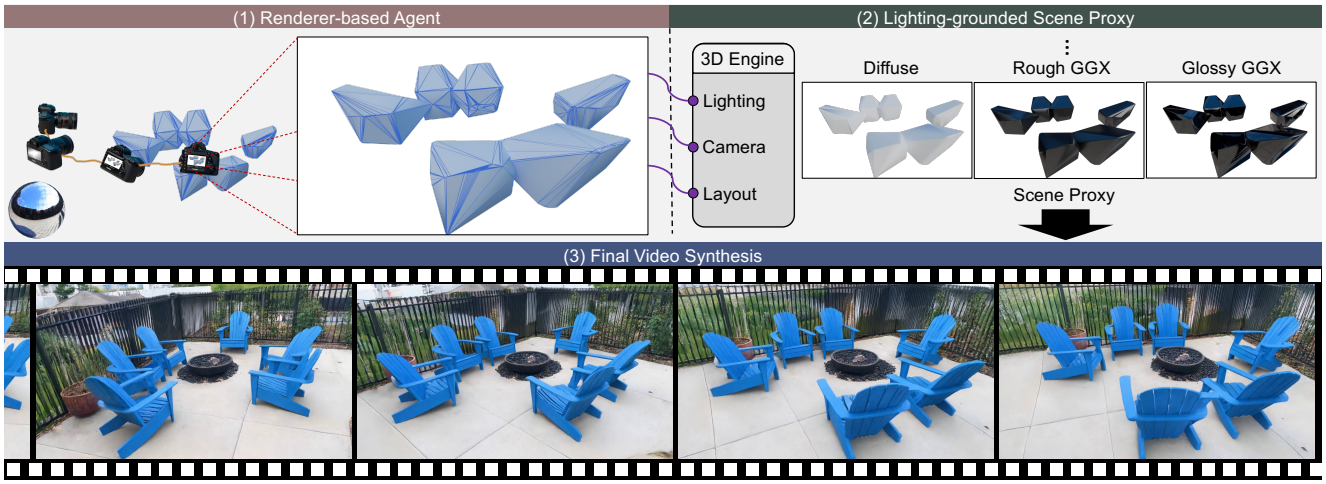

Figure 1. Overall framework. (1) A renderer-based agent produces a coarse geometric layout, camera trajectory, and a High Dynamic Range (HDR) environment map. (2) Physically-based rendering generates a lighting-grounded scene proxy containing diffuse, rough, and glossy materials with shading signals. (3) These physical cues are injected into a video diffusion model to synthesize photorealistic sequences with accurate lighting behavior, faithful scene layout, and precisely aligned camera trajectory.

Abstract

Diffusion models have achieved remarkable progress in video generation, but their controllability remains a major limitation. Key scene factors such as layout, lighting, and camera trajectory are often entangled or only weakly modeled, restricting their applicability in domains like filmmaking and virtual production where explicit scene control is essential. We present LiVER, a diffusion-based framework for scene-controllable video generation. To achieve this, we introduce a novel framework that conditions video synthesis on explicit 3D scene properties, supported by a new large-scale dataset with dense annotations of object layout, lighting, and camera parameters. Our method disentangles these properties by rendering control signals from a unified 3D representation. We propose a lightweight conditioning module and a progressive training strategy to integrate these signals into a foundational video diffusion model, ensuring stable convergence and high fidelity. Our framework enables a wide range of applications, including image-to-video and video-to-video synthesis where the underlying 3D scene is fully editable. To further enhance usability, we develop a scene agent that automatically translates high-level user instructions into the required 3D control signals. Experiments show that LiVER achieves state-of-the-art photorealism and temporal consistency while enabling precise, disentangled control over scene factors, setting a new standard for controllable video generation.

1. Introduction

Recent video generation models [4, 39, 47] have demonstrated impressive visual quality, temporal consistency, and

*Corresponding author.

diverse scenarios. Towards physically realistic video generation, researchers have paid great attention on data curation [12], prompt enhancement [50], and architecture improvement [53]. Despite these improvements, these data-driven approaches still struggle to model complex physical interactions (*e.g.*, occlusion relationships between objects in dynamic scenarios).

Introducing grounded references to explicitly model the realistic world has proven to be effective at better controllability in motion [40], layout [21, 27], and camera [11, 13, 36]. Although these 3D-aware conditions can provide a strong geometric foundation, existing works largely overlook their potential for computing physically-accurate lighting (*e.g.*, taking BRDF into consideration). Consequently, mismatching lighting effects (*e.g.*, shadows, reflections, and ambient occlusion) are still produced in realistic material representation (*e.g.*, skin, metals, and glass) in generated videos.

In this paper, we propose **LiVER**, a framework for **Li**ghting-grounded **V**ideo gen**ER**ation with renderer-based agent reasoning. As illustrated in Fig. 1, LiVER effectively generates videos with diverse and physically realistic lighting effects from text descriptions (*e.g.*, soft daylight). We first construct a renderer-based agent framework to retrieve and generate scene lighting, layout, and camera trajectory as a coarse scene representation to guide video generation. Instead of directly adopting full 3D representations, we represent the scene using a stack of 2D render passes (*e.g.*, diffuse, glossy GGX, and rough GGX) generated by a 3D engine [8]. Formulated as stacked image sequences, this proxy preserves physically realistic lighting cues from the 3D scene while providing the scene layout information. Leveraging the generative priors of a pre-trained text-to-video model (*i.e.*, Wan2.2-5B [39]), we then design a lightweight conditional encoder and adapter to achieve this alignment and translate this proxy into the final visually appealing video. Finally, to effectively optimize for these lighting effects, we propose a three-stage training scheme to enhance lighting diversity while preserving the base model's visual quality.

To facilitate model training and evaluation, we collect and render the **LiVERSet**, a **Li**ghting-grounded **V**ideo gen**ER**ation data**Set**. This dataset comprises two complementary subsets: *(i)* a real-world subset **LiVER-Real** that captures complex and realistic lighting phenomena and *(ii)* a synthetic subset **LiVER-Syn** that features diverse and controllable physically-based rendered lighting. We provide comprehensive annotations for both subsets, including scene geometry, environment maps, camera poses, and text descriptions. In total, **LiVERSet** contains over 11K videos (81 frames each at 720×1280 resolution), split into 10K for training and 1K for evaluation.

We summarize our contributions as follows:

- We propose a framework for lighting-grounded video generation with physically realistic lighting effects, and introduce a lighting-aware dataset with comprehensive annotations for both real-world and synthetic videos.
- We construct a renderer-based agent to reason a structured scene graph and render the lighting-aware scene proxy and a lightweight encoder as well as adapter to effectively align this proxy with the video latent space.
- We design a three-stage training scheme to improve lighting diversity and preserve visual quality. Extensive experiments demonstrate our method achieves state-of-the-art performance in physically-accurate lighting phenomena.

2. Related Work

2.1. Text-to-Video Generation

Recent years have witnessed remarkable progress in Text-to-Video (T2V) generation, largely driven by the success of diffusion models [15, 33]. Video foundation models [22, 39, 47] have demonstrated their ability to generate visually appealing and temporally coherent video clips from a single text description. Based on this advancement, researchers have paid attention to long video generation [17, 48] by introducing causal reasoning, improved video controllability by receiving multi-modal conditions [19, 20], and constructed multimodal datasets and benchmarks [18, 37] to evaluate shortcomings and improve performance with curated data. However, most of these approaches are data-driven, inherently struggling to model complex physical interactions for scenarios with multiple instances. We propose to model physical properties as explicit conditions to guide the model follow underlying physical principles.

2.2. 3D-grounded Video Generation

A prominent direction for improving physical realism is to ground the generation process in explicit 3D spatial information.

Early efforts used 2D proxies to improve physical realism. For instance, Boximator [42] and TrailBlazer [31] allow users to define bounding-box trajectories, while methods like Ctrl-V [30] and TrackDiffusion [24] condition generation on pre-defined object tracklets. MotionPrompting [11] employs sparse point trajectories to direct object motion. Recent models have introduced direct trajectory conditioning to model physical properties like camera. CameraCtrl [13] uses a plug-in module for explicit camera pose control, while Collaborative Video Diffusion (CVD) [23] synchronizes views along different camera paths using cross-video attention. MotionCtrl [45] provides unified control of camera motion and object motion by conditioning on camera poses and sparse object trajectories. More integrated frameworks like CineMaster [44] combine 3D box and camera conditioning.

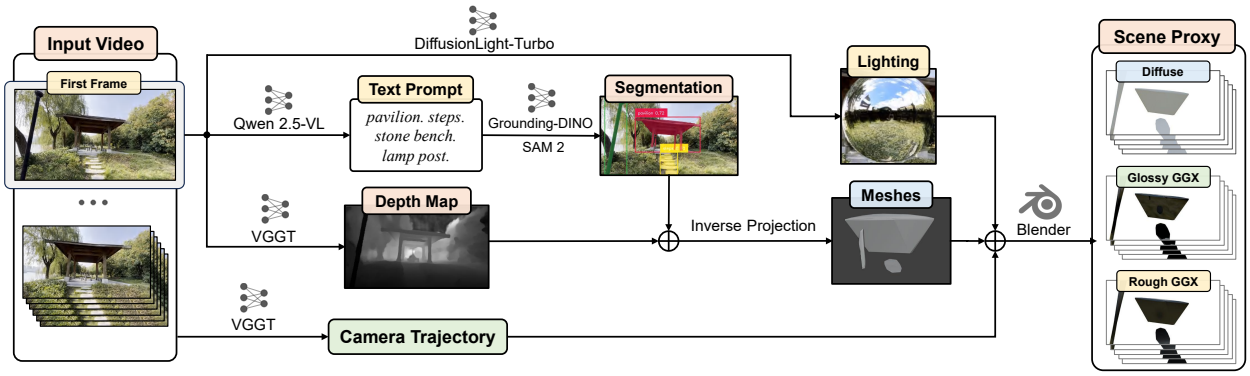

Figure 2. Our data annotation pipeline for **LiVER-Real**. We process each video to reconstruct its 3D geometry and estimate its HDR environment map. These are then used to render three pixel-aligned lighting representations (Diffuse, Glossy GGX, Rough GGX), which are concatenated to form the final conditioning input.

While these methods introduce 3D-aware conditions to provide a strong geometric foundation for video generation and improve scene consistency, they largely still ignore the physically-accurate lighting for generated videos, leading to unrealistic artifacts. To fill in this gap, we propose to model lighting as a unified part of physical properties.

2.3. Image and Video Relighting

Image and video relighting task aims to modify the illumination conditions of a scene after it has been captured, by extracting and understanding explicit lighting representations [5, 6, 32, 51].

DiLightNet [49] augments image diffusion models with explicit radiance hints for detailed lighting edits. GenLit [3] reframes single-image relighting as a video diffusion task, achieving realistic results. For video, Light-A-Video [54] uses a training-free fusion pipeline for relighting, while LumiSculpt [52] introduces a plug-in network to control light properties and motion. However, they often entangle the lighting with other physical properties like camera and scene layout.

Inspired by these works, we design our LiVER model to integrate lighting as the primary condition, rendered from the 3D scene proxy to generate general videos with physically accurate lighting while preserving controllability over scene layout and camera.

3. Dataset

We now describe the proposed **LiVERSet** in detail. The dataset contains two complementary components: *(i)* **LiVER-Real** contains real videos exhibiting complex, naturally occurring lighting together with detailed physical annotations. *(ii)* **LiVER-Syn** is a physically based rendered collection with dynamic illumination, providing broader lighting variability than what is captured in the real data. For both subsets, we derive a unified scene proxy as the primary control condition for video generation, paired with text descriptions to provide semantic guidance.

LiVER-Real Video Annotation. Existing real-world videos x^{real} lack the explicit physical annotations required for precise lighting control. Therefore, we develop an annotation pipeline to reconstruct a dynamic 3D scene and global lighting from video, as shown in Fig. 2. We first estimate per-frame camera poses c^{real} using VGGT [43], and extract only the first-frame depth map from the same model, complemented by an initial-frame object segmentation from Grounding-DINO [28] and SAM 2 [35]. This combination allows us to lift the 2D segmented objects into a coarse 3D scene mesh s^{real}. For lighting modeling, we employ DiffusionLight-Turbo [7] to estimate a single HDR environment map, which serves as a robust approximation of the scene's global lighting representation l^{real}.

LiVER-Syn Data Rendering. To supplement the diverse lighting phenomena limited in real-world data and to learn fine-grained control, we construct a complementary synthetic dataset. We first carefully curate a subset of Objaverse-XL [9] to include high-quality PBR materials. A 3D scene s^{syn} is then procedurally generated by randomly sampling several objects from this subset. To ensure the diverse lighting representation l^{syn}, we illuminate scenes using a diverse set of HDR environment maps from the Poly Haven [1] library. We then introduce dynamic effects by rotating the HDR environment map over the video's duration horizontally around the scene's vertical axis (*i.e.*, yaw rotation). The total rotation angle for each clip is sampled uniformly from the range $[180°, 240°]$ to ensure a visually significant change in lighting direction (*e.g.*, sun moves to the opposite side of the scene). By procedurally moving the camera position c^{syn}, we render the final photorealistic target video x^{syn}.

Scene Proxy Construction. The scene proxy is designed to provide realistic lighting cues by decomposing the scene's complex illumination into fundamental light-

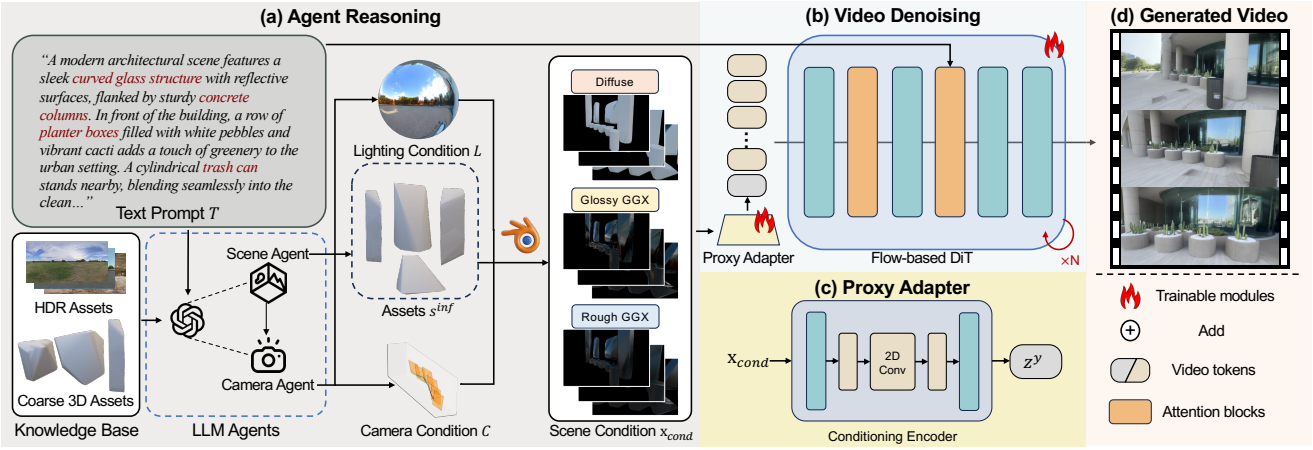

Figure 3. Pipeline of LiVER. Given a text prompt T, our Scene Agent parses object categories, spatial relations, and coarse geometry to construct an initial 3D scene. The Camera Agent infers a camera trajectory consistent with the described viewpoint and scene semantics, producing the camera condition C. The 3D scene is then rendered through a physically-based renderer to obtain the lighting-grounded scene proxy, including diffuse, glossy GGX, and rough GGX components, forming the scene condition X_{cond}. These components encode physically meaningful cues such as material response, shading, and reflections. The Lighting condition L, represented by an environment map, provides global illumination cues. The DiT-based video diffusion model integrates all conditions X_{cond}, C, L and generates a photorealistic video that preserves the scene layout, camera trajectory, and physically accurate lighting behavior.

ing components and providing scene layout information. For the two subsets, we use a physically-based renderer (Blender [8] in our implementation) to render a scene proxy $y \in \mathbb{R}^{F\times 9\times H\times W}$ based on the 3D scene mesh s^i, lighting representation l^i, and camera trajectory c^i:

$$y = [x^{\text{DIFF}}, x^{\text{GGX1}}, x^{\text{GGX2}}] = R(s^i, l^i, c^i), \qquad (1)$$

where $i \in \{\text{real}, \text{syn}\}$, F is the number of frames and R is the renderer [8]. The scene proxy is a stack of 2D render passes, including the purely diffuse x^{DIFF}, rough GGX (high roughness) x^{GGX1}, and glossy GGX (low roughness) x^{GGX2} to capture low-frequency ambient lighting, medium-frequency broad reflections, and high-frequency specular highlights, respectively.

Caption Annotation. To provide text descriptions to guide the scene semantics, we use Qwen 2.5-VL [2] as the vision-language model to generate a caption for each video clip. Example prompts used for caption generation are detailed in the supplementary material.

Dataset Statistics. Our final composite dataset comprises approximately 11K video clips, split into a 10K training set and a 1K evaluation set, both equally split between real-world and synthetic data. Each video has a duration of 81 frames at a 720×1280 resolution.

4. Methodology

In this section, we present the details of our LiVER model. We begin by introducing the renderer-based agent reasoning, which interprets a user-provided text description into an explicit scene proxy (Sec. 4.1). We then detail the approach to use this scene proxy to guide the lighting-grounded video generation, which includes the task formulation, the scene proxy encoder, and the adapter (Sec. 4.2). Finally, we present the stage-wise training scheme designed to effectively train these components, optimizing for proxy translation, lighting control, and lighting diversity (Sec. 4.3).

4.1. Renderer-based Agent Reasoning

To bridge high-level user intent with structured lighting control, we design an intelligent renderer-based agent to translate text descriptions into the scene proxy. As shown in Fig. 3, this process involves three components: scene building, lighting setup, and camera planning.

Scene Building. Given a text description, the agent first performs semantic decomposition to parse object categories and their spatial relationships, and organizes them into a structured scene graph $\mathcal{G} = (V, E)$. In this graph, each node $v_i \in V$ is an instantiated object that encapsulates semantic categories and materials properties, while an edge $e_{i,j} \in E$ represents the spatial relationships between the i-th and j-th objects (*e.g.*, "in front of"). Then, the agent retrieves a suitable mesh asset from our curated library [9] for each node v_i, and optimizes their poses to satisfy the relational constraints defined in E.

Lighting Setup. After constructing the scene geometry s^{geo} based on the scene graph, the agent parses the original text description for lighting cues (*e.g.*, "warm mood" and "overcast sky") and selects an appropriate HDR environment map from the Poly Haven library [1] to configure a physically plausible illumination setup l^{inf} that matches the described mood. When there is no suitable environment

map, the agent generates one using pretrained generation models [41].

Camera Planning. Given the static scene graph $\mathcal{G}$, the camera planner generates a dynamic camera trajectory c^{inf} for F frames. It first parses cinematographic hints from the text description (*e.g.*, "orbit", "dolly zoom", and "crane shot") to establish a camera motion plan. This defines a set of keyframe poses to specify the camera's position and orientation. A temporally smooth trajectory c^{inf} is then generated by interpolating these keyframes using a spline.

Proxy Rendering. Finally, the agent assembles reasoned assets $[s^{\text{inf}}, l^{\text{inf}}, c^{\text{inf}}]$ as the final scene representation. This representation is then fed into renderer to render the 2D scene proxy y as Eq. (1), enabling physically consistent scene authoring for lighting-grounded video generation.

4.2. Lighting-grounded Video Generation

To translate the scene proxy into videos with physically realistic lighting, our LiVER model leverages the generative priors of a pretrained video model. We additionally introduce a proxy encoder to capture lighting cues and an adapter to align proxy tokens with the video latent space.

Video Generation Backbone. We build our video generation model upon Wan2.2-5B [39] to leverage its priors for realistic video generation. We use a spatiotemporal Variational Autoencoder (VAE) to map high-dimensional videos x into a compact latent space as $z = \mathcal{E}(x)$. During training, we follow the flow matching formulation [26] to sample a Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$. This noise is then linearly interpolated with the latent code $z_t = tz + (1-t)\epsilon$ for a random timestep $t \in [0, 1]$. The model is trained to predict the ground truth velocity vector $v_t = \frac{\mathrm{d}z_t}{\mathrm{d}t} = z - \epsilon$:

$$\mathcal{L} = \mathbb{E}_{z,\epsilon,t}\left[\left|u_\theta(z_t, y, c^{\text{txt}}, t) - v_t\right|^2\right], \tag{2}$$

where y is the scene proxy, c^{txt} is the text embedding of the caption, and u_θ is our LiVER model.

Scene Proxy Encoding. As the scene proxy is organized into a stack of 2D render passes $y \in \mathbb{R}^{F\times 9\times H\times W}$, we can simply employ a lightweight 2D proxy encoder $\mathcal{E}_{\text{proxy}}$ to map it into a compact feature representation. Architecturally, the proxy encoder is implemented by multiple 2D convolutional blocks, each containing a convolution, a GroupNorm [46] layer, and a SiLU [10] non-linearity. The network progressively downsamples the spatial dimensions while mapping the input to higher-dimensional features $z^{\text{y}} = \mathcal{E}_{\text{proxy}}(y)$ for each frame, where $z^{y} \in \mathbb{R}^{F\times C\times H'\times W'}$, $H', W' = H/16, W/16$ are the downsampled resolution, and C is the dimensions of the features. This design minimizes computational overhead while capturing lighting cues and providing the information about the scene.

Video Latent Integration. To ensure the video latent code semantically aligns with the scene proxy, we design a lightweight conditioning encoder to inject the encoded scene proxy features z^{y} directly into the video latent space. Specifically, we first stack the multiple RGB rendered images along the channel dimension to form a 9-channel input. Instead of using complex 3D convolutions, we employ a 2D convolutional network with a sequence of downsampling blocks to extract spatial features $z^{\text{y}} \in \mathbb{R}^{F\times C\times H'\times W'}$. This explicitly aligns the spatial resolution and channel dimensions of the proxy features with the VAE latent code $z \in \mathbb{R}^{C\times H'\times W'}$ of the video, where $C = 4$. To guide the video generation direction, the encoder learns a spatial residual that is directly superimposed onto the video latent code. These encoded proxy features z^{y} are used to modulate the original video latents z:

$$z' = z + \alpha \cdot z^{\text{y}}, \tag{3}$$

where z^{y} is the output of the 2D conditioning encoder, and α is a learnable scalar weight initialized to zero. This strategy ensures the conditioning encoder has no initial impact on video generation at the start of training, allowing the proxy features to gradually guide the video latent space and ultimately enable lighting-grounded precise control.

4.3. Stage-wise Training Scheme

To effectively learn the conditioning pathway while preserving the video backbone's generative priors, we adopt a three-stage training scheme to optimize for proxy translation, lighting control, and lighting diversity.

Conditional Pathway Training. We first freeze the entire video diffusion backbone and train only the proxy encoder and adapter modules for 10 epochs. This initial stage aims to translate the scene proxy into coarse control signals over the generation process.

Joint LoRA Fine-tuning. We unfreeze LoRA [16] layers integrated into the video backbone, and jointly fine-tune these layers along with the proxy encoder and adapter for another 10 epochs. This stage refines the semantic alignment, effectively balancing proxy controllability and overall visual quality.

Lighting Diversity Expansion. We continue the joint LoRA fine-tuning while mixing real videos with our synthetic data in a $1:1$ ratio. This final stage adapts the model to more general scenarios and enhances its ability to render diverse lighting phenomena.

5. Experiments

5.1. Implementation Details

Our model is built upon the Wan 2.2-5B-TI2V checkpoint [39], inheriting its robust generative priors. We use

LoRA [16] to reduce computational costs and prevent catastrophic forgetting of the base model's capabilities.

Training is conducted on our curated dataset for approximately 100K steps. We use 8 NVIDIA H100 GPUs with a per-GPU batch size of 2, resulting in a total batch size of 16. We employ the AdamW optimizer [29] with a constant learning rate of 1×10^{-5}. The model generates videos at a resolution of 704×1280. Further details on the network architecture and training hyperparameters are provided in the supplementary material.

5.2. Baselines

We conduct a comprehensive quantitative and qualitative comparison against several state-of-the-art methods capable of generating video from 3D-aware conditions:

- CameraCtrl [13]: Controls video generation by conditioning on explicit camera pose sequences to enforce camera-consistent motion.
- MotionCtrl [45]: Controls both camera and object motion by conditioning the diffusion model on camera poses and sparse object trajectories through lightweight temporal and spatial motion modules.
- VideoFrom3D [21]: Generates 3D scene videos from coarse geometry by producing anchor views with an image diffusion model and interpolating them via a video diffusion model.

For a fair comparison, all methods are evaluated on a held-out test set derived from our dataset. Since CameraCtrl and MotionCtrl only generate 16-frame clips, we compare ours against these methods using the first 16 frames of our generated videos. For VideoFrom3D, comparisons follow our 81-frame evaluation protocol, and thus both methods are evaluated using full 81 frames. Since VideoFrom3D requires training a style-specific LoRA for each test sample ($\sim$ 40 minutes on an NVIDIA H100 GPU), we train and evaluate VideoFrom3D on only 20 videos randomly selected from our test set (*i.e.*, 20 LoRAs). Consequently, the quantitative results reported for VideoFrom3D are based on this 20-sample subset.

5.3. Evaluation Metrics

We assess performance using a suite of standard metrics targeting different aspects of video generation:

Quality & Realism. We use Fréchet Video Distance (FVD) [38] to assess distributional similarity between generated and real videos, and per-frame Fréchet Inception Distance (FID) [14] to evaluate image quality. For semantic consistency with the text prompts, we report CLIP image–text similarity computed with the pretrained CLIP model [34].

Control Fidelity. To evaluate how faithfully the models adhere to the input conditions, we measure:

- **Camera Pose Error:** We estimate the camera poses from the generated videos using VGGT [43], align each predicted trajectory to the ground truth via a global Sim(3), and report absolute trajectory error (ATE), mean per-step translation error (RPEt) and mean per-step rotation error (RPEr). For readability, ATE and RPEt are scaled by $\times 100$, while RPEr is in degrees.
- **Lighting Error:** We use an existing lighting estimator [7] to recover HDR environment maps from the generated video frames. We then compute the scale-invariant mean squared error (SI-MSE) between the predicted and ground-truth lighting. This metric provides both an overall lighting error and a lighting instability measure, defined as the standard deviation of the SI-MSE over time.
- **Layout Error:** Frame-wise object masks are obtained via a segmentation model [35], and compared to ground-truth masks using mean Intersection-over-Union (mIoU) to assess how accurately the generated videos preserve scene layout and object shapes.

5.4. Quantitative Comparison

We evaluate our approach against publicly available baseline models on the test split of LiVER-Real, which is composed of original videos sourced from the public dataset of Ling et al. [25]. Table 1 summarizes the quantitative results. Our method attains the lowest FVD and FID scores and the highest CLIP score, indicating improvements in both video quality and overall realism. In addition, our model achieves the highest control fidelity, exhibiting reduced camera pose and lighting errors. It also delivers the highest mIoU, demonstrating more accurate preservation of object shapes and spatial structure throughout the sequence.

5.5. Qualitative Evaluation

We present a visual comparison of our method with the baseline approaches in Fig. 4. As shown, our method achieves more realistic lighting effects as well as more precise layout and camera control, resulting in outputs that most closely match the reference video (Ref).

5.6. Ablation Study

Impact of Synthetic Data. To validate the importance of our synthetic dataset, we trained a variant of our model exclusively on real-world videos. As shown in Fig. 6 (first column), this model produces wrong and mostly even illumination, failing to reproduce correct lighting effects. This result confirms that the diverse and dynamic illumination in our LiVER-Syn data is crucial for enabling lighting control and preventing the model from overfitting to the less varied lighting patterns typical of real-world footage.

Importance of the Staged Training Scheme. We also evaluated our multi-stage training strategy by training a

Table 1. Quantitative comparison with state-of-the-art methods. Our method consistently outperforms the baselines. †Only compare first 16 frames.

Method	FVD ↓	FID ↓	CLIP ↑	ATE ↓	RPEt ↓	RPEr ↓	LE ↓	LI ↓	mIoU ↑
CameraCtrl [13]	48.03	98.29	28.75	2.15	1.39	1.68	0.06	0.03	0.68
MotionCtrl [45]	63.13	97.21	26.67	3.42	2.03	7.32	0.07	0.04	0.66
Ours†	**32.45**	**42.32**	**29.62**	**1.30**	**0.81**	**1.16**	**0.05**	**0.02**	**0.86**
VideoFrom3D [21]	36.94	157.89	24.51	17.55	3.85	3.12	0.05	0.03	0.74
Ours	**32.56**	**129.56**	**30.97**	**2.48**	**0.71**	**0.50**	**0.04**	**0.02**	**0.87**

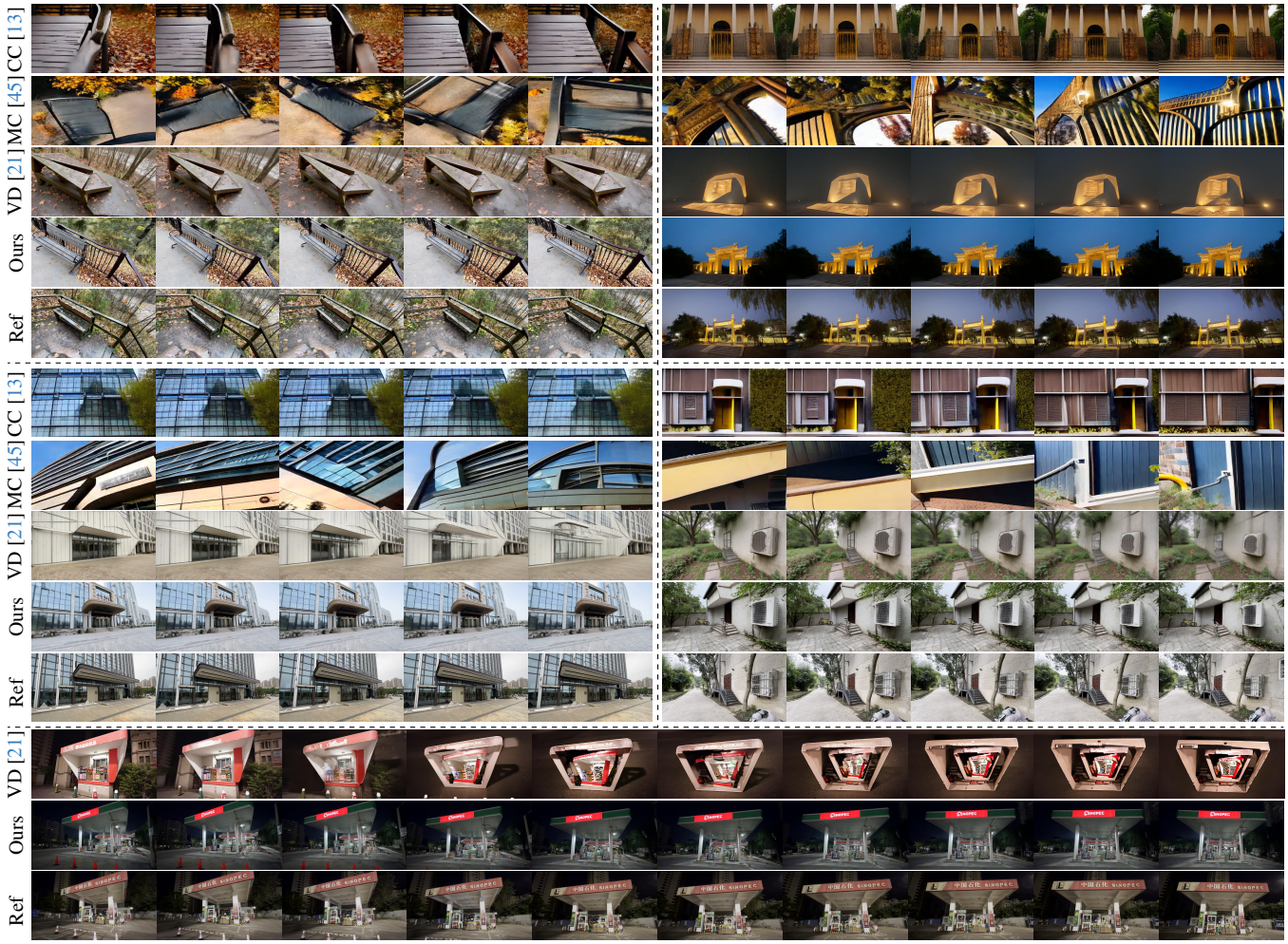

Figure 4. Qualitative comparison with state-of-the-art controllable video generation models. In each block, each row corresponds to one video, and frames are arranged from left to right in temporal order. The top row shows the results of each comparison method, followed by ours, with the ground truth (GT) shown in the final row.

variant end-to-end on the full dataset from scratch. As visualized in Fig. 6 (second column), this joint training approach leads to nearly still output and a notable degradation in performance. The model's ability to precisely follow specified scene conditions is diminished. This suggests simultaneously learning control signals while adapting a large-scale pretrained model presents a more challenging optimization problem. Our staged approach, which progressively introduces the conditioning, proves essential for ensuring stable convergence and effectively integrating our control module without corrupting the generative priors.

5.7. User Study

To further assess the perceptual quality and consistency of our results, we conduct a user study involving 25 participants. Each participant is shown 20 sets of videos, where each set contains results from four competing methods applied to the same underlying scene. For every set, participants are asked to select, for each evaluation dimension, the method they prefer the most: *video quality (VQ)*, *scene control (SC)*, *camera trajectory control (CC)*, and *lighting control (LC)*. We count, for each method and each dimen-

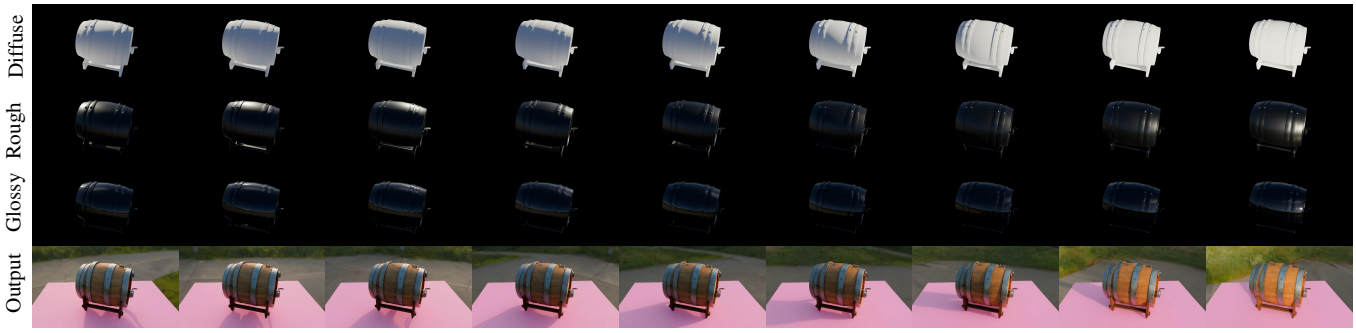

Figure 5. By manipulating the HDR environment map, our model produces continuous and physically consistent lighting variations. We show the diffuse, glossy, and rough GGX components of the scene proxy (top three rows) and the corresponding synthesized outputs (bottom). Lighting changes are reflected in shading and reflections while geometry and materials remain stable.

w/o SynData w/o staged Ours GT

Figure 6. Qualitative results of our ablation study.

Table 2. Percentage of samples in which each method is selected as the most preferred solution.

Method	VQ ↑	SC ↑	CC ↑	LC ↑
CameraCtrl [13]	4.3%	4.1%	2.2%	6.0%
MotionCtrl [45]	3.6%	3.6%	1.6%	5.7%
VideoFrom3D [21]	8.7%	8.9%	24.1%	29.0%
Ours	**83.4%**	**83.3%**	**72.1%**	**59.3%**

sion, the percentage of samples in which it is chosen as the preferred solution. As summarized in Table 2, our method outperforms across all four dimensions.

5.8. Controllability

Lighting Control. Our method provides fine-grained control over illumination. Shown in Fig. 5, by manipulating the HDR environment map, our model generates videos with dynamic and continuous lighting changes, such as rotating lighting. This is achieved while maintaining the consistency of the scene's geometry and material properties, showcasing a good disentanglement of lighting from scene structure.

Layout and Camera Control. The use of an explicit 3D scene proxy as conditioning ensures geometrically precise control over both object layout and camera motion. As qualitatively and quantitatively validated in Fig. 4 and Tab. 1, our method exhibits superior performance in accurately placing objects and following user-defined camera trajectories compared to methods relying on 2D-based proxies.

Flexible Editing Workflow. A key design of our method is to empower users with a flexible and intuitive workflow. We introduce a renderer-based agent that automatically generates an initial 3D proxy, simplifying the creation of conditioning signals. Crucially, this proxy is not a fixed input but a fully editable starting point. Users can import it into standard 3D software to perform traditional edits: they can add, delete, or move geometry, refine the lighting conditions, and design new camera trajectory. This hybrid approach uniquely combines automated scene setup with the creative freedom of established CGI pipelines.

6. Conclusion

We present LiVER, a novel diffusion-based framework for controllable video generation that jointly models layout, lighting, and camera. Unlike prior work that focuses solely on text prompts or global scene cues, LiVER offers explicit, fine-grained control over the spatiotemporal composition of generated content. Our approach bridges the gap between generative quality and scene-level controllability, paving the way for practical deployment in creative media, virtual cinematography, and immersive content production.

Limitations. As our initial 3D reconstruction of the scene geometry is coarse, the model relies on the text description to synthesize fine-grained geometric and material details. This makes the final output quality (*e.g.*, geometric consistency) sensitive to the user-provided prompts. We will explore improving the agent's scene interpretation through more sophisticated prompt engineering in our future work to mitigate this issue.

Acknowledgment

This work is supported by National Natural Science Foundation of China (Grant No. 62136001) and Beijing Major Science and Technology Project (Grant No. Z251100008125009). PKU-affiliated authors thank openbayes.com for providing computing resources.

References

[1] Poly Haven: https://polyhaven.com. 3, 4

[2] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025. 4

[3] Shrisha Bharadwaj, Haiwen Feng, Giorgio Becherini, Victoria Fernandez Abrevaya, and Michael J Black. GenLit: Reformulating single-image relighting as video generation. *arXiv preprint arXiv:2412.11224*, 2024. 3

[4] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. 1

[5] Ziqi Cai, Kaiwen Jiang, Shu-Yu Chen, Yu-Kun Lai, Hongbo Fu, Boxin Shi, and Lin Gao. Real-time 3D-aware portrait video relighting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 3

[6] Ziqi Cai, Shuchen Weng, Yifei Xia, and Boxin Shi. PhyS-EdiT: Physics-aware semantic image editing with text description. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. 3

[7] Worameth Chinchuthakun, Pakkapon Phongthawee, Nontaphat Sinsunthithet, Amit Raj, Varun Jampani, Pramook Khungurn, and Supasorn Suwajanakorn. DiffusionLight-Turbo: Accelerated light probes for free via single-pass chrome ball inpainting. *arXiv preprint arXiv:2507.01305*, 2025. 3, 6

[8] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. 2, 4

[9] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, Eli VanderBilt, Aniruddha Kembhavi, Carl Vondrick, Georgia Gkioxari, Kiana Ehsani, Ludwig Schmidt, and Ali Farhadi. Objaverse-XL: A universe of 10M+ 3D objects. In *Advances in Neural Information Processing Systems*, 2023. 3, 4

[10] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural networks*, 2018. 5

[11] Daniel Geng, Charles Herrmann, Junhwa Hur, Forrester Cole, Serena Zhang, Tobias Pfaff, Tatiana Lopez-Guevara, Yusuf Aytar, Michael Rubinstein, Chen Sun, et al. Motion prompting: Controlling video generation with motion trajectories. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. 2

[12] Jing Gu, Xian Liu, Yu Zeng, Ashwin Nagarajan, Fangrui Zhu, Daniel Hong, Yue Fan, Qianqi Yan, Kaiwen Zhou, Ming-Yu Liu, et al. "PhyWorldBench": A comprehensive evaluation of physical realism in text-to-video models. *arXiv preprint arXiv:2507.13428*, 2025. 2

[13] Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. CameraCtrl: Enabling camera control for video diffusion models. In *International Conference on Learning Representations*, 2025. 2, 6, 7, 8

[14] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, 2017. 6

[15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020. 2

[16] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. 5, 6

[17] Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self Forcing: Bridging the train-test gap in autoregressive video diffusion. *arXiv preprint arXiv:2506.08009*, 2025. 2

[18] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 2

[19] Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. Vace: All-in-one video creation and editing. *arXiv preprint arXiv:2503.07598*, 2025. 2

[20] Xuan Ju, Weicai Ye, Quande Liu, Qiulin Wang, Xintao Wang, Pengfei Wan, Di Zhang, Kun Gai, and Qiang Xu. FullDit: Multi-task video generative foundation model with full attention. *arXiv preprint arXiv:2503.19907*, 2025. 2

[21] Geonung Kim, Janghyeok Han, and Sunghyun Cho. VideoFrom3D: 3D scene video generation via complementary image and video diffusion models. In *ACM SIGGRAPH Asia Conference Papers*, 2025. 2, 6, 7, 8

[22] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. HunyuanVideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024. 2

[23] Zhengfei Kuang, Shengqu Cai, Hao He, Yinghao Xu, Hongsheng Li, Leonidas J Guibas, and Gordon Wetzstein. Collaborative video diffusion: Consistent multi-video generation with camera control. In *Advances in Neural Information Processing Systems*, 2024. 2

[24] Pengxiang Li, Kai Chen, Zhili Liu, Ruiyuan Gao, Lanqing Hong, Dit-Yan Yeung, Huchuan Lu, and Xu Jia. TrackDiffusion: Tracklet-conditioned video generation via diffusion models. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2025. 2

[25] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu,

et al. DL3DV-10K: A large-scale scene dataset for deep learning-based 3D vision. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 6

[26] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. 5

[27] Feng-Lin Liu, Shi-Yang Li, Yan-Pei Cao, Hongbo Fu, and Lin Gao. Sketch3DVE: Sketch-based 3D-aware scene video editing. In *ACM SIGGRAPH Conference Papers*, 2025. 2

[28] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. In *European Conference on Computer Vision*, 2023. 3

[29] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 6

[30] Ge Ya Luo, ZhiHao Luo, Anthony Gosselin, Alexia Jolicoeur-Martineau, and Christopher Pal. Ctrl-V: Higher fidelity autonomous vehicle video generation with bounding-box controlled object motion. *Transactions on Machine Learning Research*, 2025. 2

[31] Wan-Duo Kurt Ma, John P Lewis, and W Bastiaan Kleijn. Trailblazer: Trajectory control for diffusion-based video generation. In *ACM SIGGRAPH Conference Papers*, 2024. 2

[32] Rohit Pandey, Sergio Orts-Escolano, Chloe Legendre, Christian Haene, Sofien Bouaziz, Christoph Rhemann, Paul E Debevec, and Sean Ryan Fanello. Total Relighting: learning to relight portraits for background replacement. *ACM SIGGRAPH Conference Papers*, 2021. 3

[33] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *International Conference on Computer Vision*, 2023. 2

[34] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021. 6

[35] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. SAM 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 3, 6

[36] Xuanchi Ren, Tianchang Shen, Jiahui Huang, Huan Ling, Yifan Lu, Merlin Nimier-David, Thomas Müller, Alexander Keller, Sanja Fidler, and Jun Gao. GEN3C: 3D-informed world-consistent video generation with precise camera control. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. 2

[37] Kaiyue Sun, Kaiyi Huang, Xian Liu, Yue Wu, Zihan Xu, Zhenguo Li, and Xihui Liu. T2V-CompBench: A comprehensive benchmark for compositional text-to-video generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. 2

[38] Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018. 6

[39] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. 1, 2, 5

[40] Chen Wang, Chuhao Chen, Yiming Huang, Zhiyang Dou, Yuan Liu, Jiatao Gu, and Lingjie Liu. PhysCtrl: Generative physics for controllable and physics-grounded video generation. *arXiv preprint arXiv:2509.20358*, 2025. 2

[41] Guangcong Wang, Yinuo Yang, Chen Change Loy, and Ziwei Liu. StyleLight: HDR panorama generation for lighting estimation and editing. In *European Conference on Computer Vision*, 2022. 5

[42] Jiawei Wang, Yuchen Zhang, Jiaxin Zou, Yan Zeng, Guoqiang Wei, Liping Yuan, and Hang Li. Boximator: Generating rich and controllable motions for video synthesis. *arXiv preprint arXiv:2402.01566*, 2024. 2

[43] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. 3, 6

[44] Qinghe Wang, Yawen Luo, Xiaoyu Shi, Xu Jia, Huchuan Lu, Tianfan Xue, Xintao Wang, Pengfei Wan, Di Zhang, and Kun Gai. Cinemaster: A 3D-aware and controllable framework for cinematic text-to-video generation. In *ACM SIGGRAPH Conference Papers*, 2025. 2

[45] Zhouxia Wang, Ziyang Yuan, Xintao Wang, Yaowei Li, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. MotionCtrl: A unified and flexible motion controller for video generation. In *ACM SIGGRAPH Conference Papers*, 2024. 2, 6, 7, 8

[46] Yuxin Wu and Kaiming He. Group normalization. In *European Conference on Computer Vision*, 2018. 5

[47] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. CogVideoX: Text-to-video diffusion models with an expert transformer. In *International Conference on Learning Representations*, 2025. 1, 2

[48] Tianwei Yin, Qiang Zhang, Richard Zhang, William T Freeman, Fredo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast autoregressive video diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. 2

[49] Chong Zeng, Yue Dong, Pieter Peers, Youkang Kong, Hongzhi Wu, and Xin Tong. DiLightNet: Fine-grained lighting control for diffusion-based image generation. In *ACM SIGGRAPH Conference Papers*, 2024. 3

[50] Ke Zhang, Cihan Xiao, Yiqun Mei, Jiacong Xu, and Vishal M Patel. Think Before You Diffuse: LLMs-guided physics-aware video generation. *arXiv preprint arXiv:2505.21653*, 2025. 2

[51] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Scaling in-the-wild training for diffusion-based illumination harmonization and editing by imposing consistent light transport. In *International Conference on Learning Representations*, 2025. 3

[52] Yuxin Zhang, Dandan Zheng, Biao Gong, Jingdong Chen, Ming Yang, Weiming Dong, and Changsheng Xu. Lumisculpt: A consistency lighting control network for video generation. *arXiv preprint arXiv:2410.22979*, 2024. 3

[53] Yifu Zhang, Hao Yang, Yuqi Zhang, Yifei Hu, Fengda Zhu, Chuang Lin, Xiaofeng Mei, Yi Jiang, Bingyue Peng, and Zehuan Yuan. Waver: Wave your way to lifelike video generation. *arXiv preprint arXiv:2508.15761*, 2025. 2

[54] Yujie Zhou, Jiazi Bu, Pengyang Ling, Pan Zhang, Tong Wu, Qidong Huang, Jinsong Li, Xiaoyi Dong, Yuhang Zang, Yuhang Cao, et al. Light-a-video: Training-free video relighting via progressive light fusion. In *International Conference on Computer Vision*, 2025. 3

Lighting-grounded Video Generation with Renderer-based Agent Reasoning

Supplementary Material

Ziqi Cai[1,2,4] Taoyu Yang[1,2] Zheng Chang[5] Si Li[5] Han Jiang[4] Shuchen Weng[3,1] Boxin Shi[1,2,*]

[1]State Key Laboratory for Multimedia Information Processing, School of Computer Science, Peking University
[2]National Engineering Research Center of Visual Technology, School of Computer Science, Peking University
[3]Beijing Academy of Artificial Intelligence [4]OpenBayes Information Technology Co., Ltd.
[5]School of Artificial Intelligence, Beijing University of Posts and Telecommunications

{czq,yangty1031}@stu.pku.edu.cn, {zhengchang98,lisi}@bupt.edu.cn,
hahn@openbayes.com, {shuchenweng,shiboxin}@pku.edu.cn

A. Additional Results

Due to space limitations in the main paper, we present additional experimental results in this section, including dynamic subject generation, image-to-video generation, and video-to-video generation. These results further demonstrate the controllability and robustness of our method.

A.1. Dynamic Subject Generation

Since most of our comparison methods [4, 6] are primarily evaluated on static scenes, we present static scenes in the main paper for fairness and direct comparison. However, our method is not limited to these scene types. Our renderer-based agent can render dynamic subjects, therefore enabling our model to generate dynamic scenes with physically realistic results. We present these cases in Fig. A.

A.2. Image-to-Video Generation

Our LiVER model inherently supports image-to-video generation. After reconstructing 3D meshes from a single input image, our renderer-based agent plans the camera trajectory and lighting conditions according to a user-provided text description, thereby constructing a lighting-grounded scene proxy that guides the video synthesis process. We demonstrate this capability in Fig. B, showcasing the same scene layout under two different camera trajectories and three user prompts. The top three rows correspond to the first camera trajectory, with each row reflecting a distinct user prompt; the bottom three rows show results from an alternative camera trajectory.

A.3. Video-to-Video Generation

Our LiVER model also supports video-to-video generation. We can fully extract the lighting-aware scene proxy from the reference video, and synthesize the edited video based on this proxy and user-provided text descriptions. We present this application in Fig. C, where the lighting, scene layout, and camera trajectory are well preserved.

A.4. Failure Cases

We present failure cases in Fig. D to highlight potential areas for improvement. In the first row, our model correctly renders the scene layout and follows the camera trajectory. However, the car is incorrectly depicted as having two rear ends and no discernible front. In the second row, our model fails to correctly render the details of the table tennis set. These issues can primarily be attributed to the semantic limitations of the underlying video generation backbone. We expect that further scaling up the model capacity would mitigate such errors.

B. Dataset Details

B.1. Dataset Sample Visualization

As illustrated in Sec. 3, we collect **LiVERSet** to facilitate model training and evaluation. We separately showcase three randomly selected samples from the real-world subset **LiVER-Real** and the synthetic subset **LiVER-Syn** in Fig. E to demonstrate the diversity and scope of our data.

B.2. Dataset Caption Annotation

We utilize Qwen2.5-VL-32B-Instruct [1] to generate captions from full video sequences. The model is prompted to produce concise and semantically rich paragraphs detailing scene content and object interactions. Notably, we explicitly exclude camera motion and lighting information to ensure the captions focus mainly on scene semantics, decoupling them from the proxy's physical control signals. These captions serve as text descriptions for the video backbone. The specific prompt is detailed below:

```
Task: You are a video caption generator.
Produce one concise paragraph caption suitable
 for a video-generation model (wan2.2).
Rules:
- You should NOT describe camera movements or
angles.
```

Figure A. Additional qualitative results for dynamic subject generation.

Figure B. Additional qualitative results for image-to-video generation.

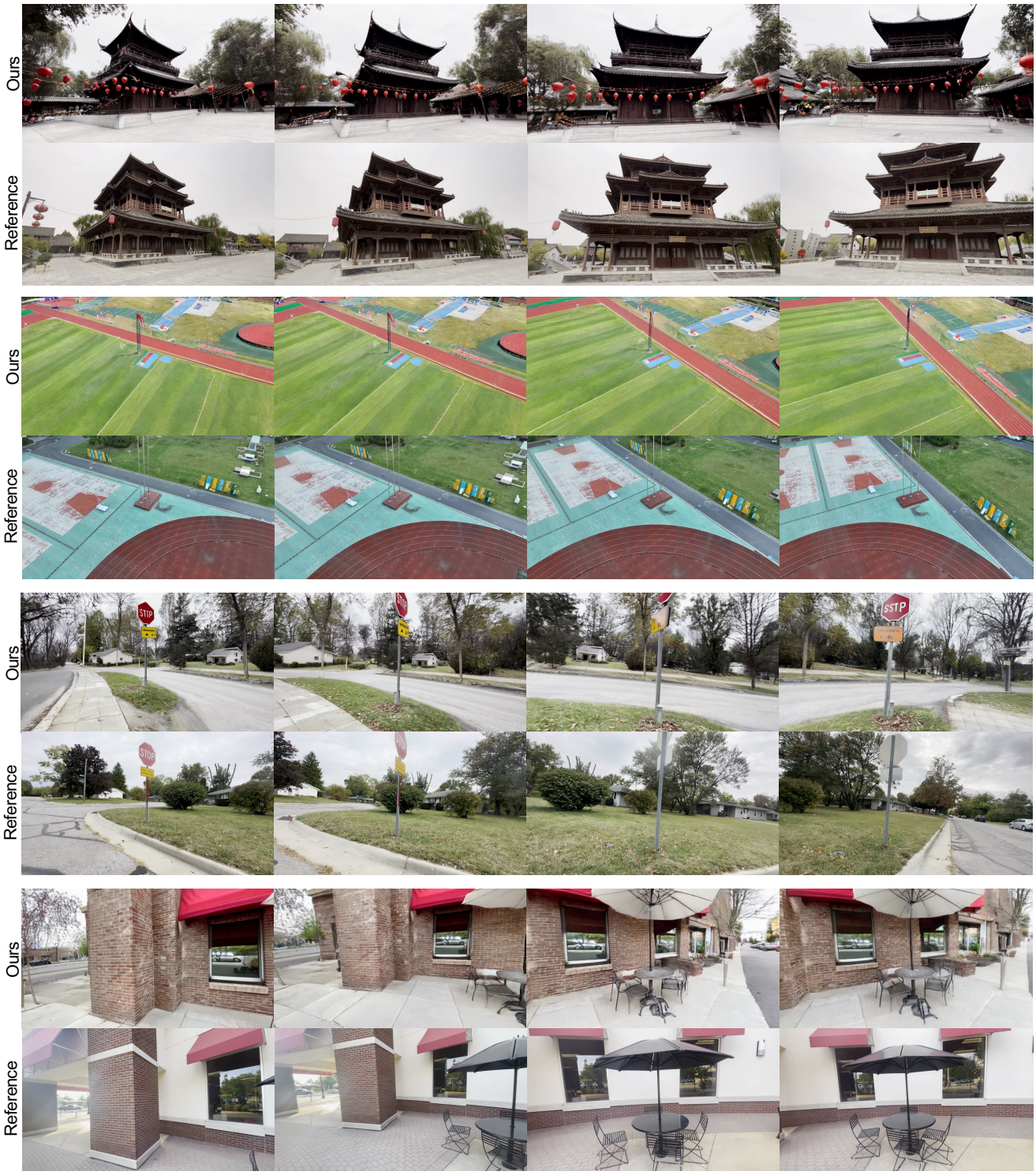

Figure C. Additional qualitative results for video-to-video generation.

Figure D. Visualization of failure cases, primarily attributed to the video backbone's semantic limitation.

```
- You should NOT include any lighting
information.
Constraints:
- Keep the caption concise and actionable for
a video generator (around 1 paragraph, up to
~100 words).
- Output exactly one paragraph (no extra lines
).
Generate caption:
```

C. Experiment Details

C.1. Baselines

In this section, we detail the configurations for all baseline methods. To ensure a fair comparison, all models are evaluated using the same prompts, scene specifications, and ground-truth camera trajectories unless otherwise noted.

CameraCtrl [4] enables explicit 6-DoF camera pose control by accepting per-frame extrinsics as input. As the standard model is limited to generating 16-frame sequences, we restrict our quantitative comparison to the first 16 frames of the ground-truth trajectories to match its sequence length.

MotionCtrl [13] accepts 6-DoF camera extrinsics and controls object motion via sparse trajectories. Similar to CameraCtrl, this method generates 16-frame videos. We adopt the same evaluation protocol, comparing the model's output against the first 16 frames of our reference data.

VideoFrom3D [6] takes 3D geometry, a camera trajectory, and a style image as input. Unlike the explicit per-frame conditioning of the other baselines, it employs a two-stage pipeline: first generating geometric anchor frames (start, middle, and end) via an image diffusion model, followed by frame interpolation using video diffusion. Following the official implementation, we synthesize the sequence and uniformly sample 81 frames from the 92-frame output to match our evaluation setting. Due to the significant computational cost ($\sim$ 40 min on a H100 GPU) of training a style-specific LoRA for every sample, we evaluate this method on a subset of 20 scenes.

C.2. Evaluation Details

Due to the varying generation capabilities of the baselines, we employ two evaluation protocols. For comparisons involving CameraCtrl and MotionCtrl (limited to 16 frames), we compute metrics on the first 16 frames of the generated and ground-truth sequences. For comparisons involving VideoFrom3D, we use the full 81-frame sequences on a representative 20-video subset. During evaluation, we utilize nine quantitative metrics to evaluate LiVER's performance across six aspects:

Frame Quality (FID). To measure the quality of each video frame, we calculate Fréchet Inception Distance (FID) [5] with standard Inception-V3 features [10]. To ensure a fair

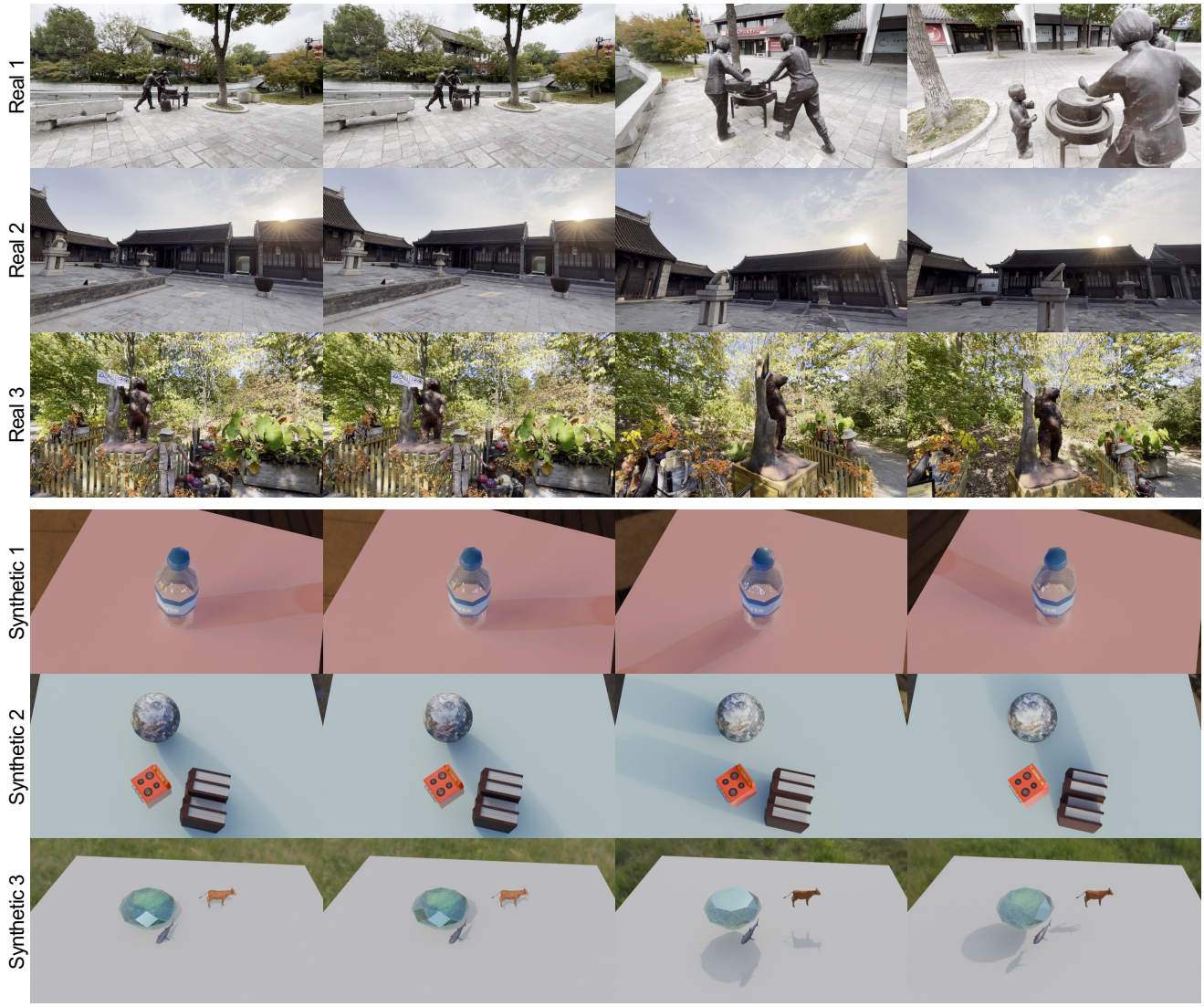

Figure E. Dataset samples. **Top:** Samples from LiVER-Real capture real-world scenes with complex, naturally occurring illumination. **Bottom:** Samples from LiVER-Syn provide physically based, controllable lighting variations.

comparison across methods with differing native resolutions, we resize all frames to 299×299 pixels. These standardized frames are processed via `pytorch-fid` [9] to compute scores on the test subsets.

Video Quality. We evaluate temporal coherence and video realism using Fréchet Video Distance (FVD) [11]. Frames are resized to a 256-pixel short side, center-cropped to 224×224, and normalized using Kinetics statistics. These inputs are fed into a pretrained I3D network [2] to extract features. We report the Fréchet distance between the generated and real video features.

Image–Text Similarity. We assess semantic alignment using OpenCLIP ViT-B/32 [7]. For each video, we uniformly subsample frames and compute the cosine similarity between their embeddings and the corresponding captions. We report the CLIP Score by calculating the average similarity across the dataset.

Trajectory Error. We evaluate camera control precision by estimating trajectories from generated videos using VGGT [12]. The estimated trajectories are aligned to the ground truth via Sim(3) Umeyama alignment on camera centers. We report three error metrics: the root mean squared Absolute Trajectory Error (ATE), the translational Relative Pose Error (RPE_t), and the rotational Relative Pose Error (RPE_r).

Lighting Error. We measure illumination consistency via

a reverse-rendering approach. Using a lighting estimator [3], we derive HDR environment maps from the generated frames and compare them to the ground truth using scale-invariant mean squared error. We report its mean as the overall Lighting Error (LE) and the temporal standard deviation to quantify Lighting Instability (LI).

Layout Accuracy. To quantify spatial alignment with the input guidance, we employ a segmentation model [8] to extract subject masks from the generated videos. We calculate the mean Intersection-over-Union (mIoU) between these predicted masks and the ground-truth instance masks; a higher mIoU indicates superior adherence to the specified scene layout.

D. Organization of Supplementary Video

We provide a supplementary video to dynamically showcase our generation results. The video is structured as follows: *(i)* **Scene proxy rendering.** We visualize the real-world video annotation pipeline to construct the scene proxy. *(ii)* **Representative video results.** We demonstrate the video generation results conditioned on the scene proxy, and present the diverse application scenarios. *(iii)* **Comparison with baselines.** We showcase comparisons with baseline methods [4, 6, 13], followed by ablation studies. *(iv)* **Failure cases.** We finally include failure cases to show potential areas for improvement.

References

[1] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[2] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

[3] Worameth Chinchuthakun, Pakkapon Phongthawee, Nontaphat Sinsunthithet, Amit Raj, Varun Jampani, Pramook Khungurn, and Supasorn Suwajanakorn. DiffusionLight-Turbo: Accelerated light probes for free via single-pass chrome ball inpainting. *arXiv preprint arXiv:2507.01305*, 2025.

[4] Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. CameraCtrl: Enabling camera control for video diffusion models. In *International Conference on Learning Representations*, 2025.

[5] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, 2017.

[6] Geonung Kim, Janghyeok Han, and Sunghyun Cho. VideoFrom3D: 3D scene video generation via complementary image and video diffusion models. In *ACM SIGGRAPH Asia Conference Papers*, 2025.

[7] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.

[8] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. SAM 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

[9] Maximilian Seitzer. pytorch-fid: FID Score for PyTorch. `https://github.com/mseitzer/pytorch-fid`, 2020. Version 0.3.0.

[10] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

[11] Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018.

[12] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[13] Zhouxia Wang, Ziyang Yuan, Xintao Wang, Yaowei Li, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. MotionCtrl: A unified and flexible motion controller for video generation. In *ACM SIGGRAPH Conference Papers*, 2024.